6.0 CONTROLS

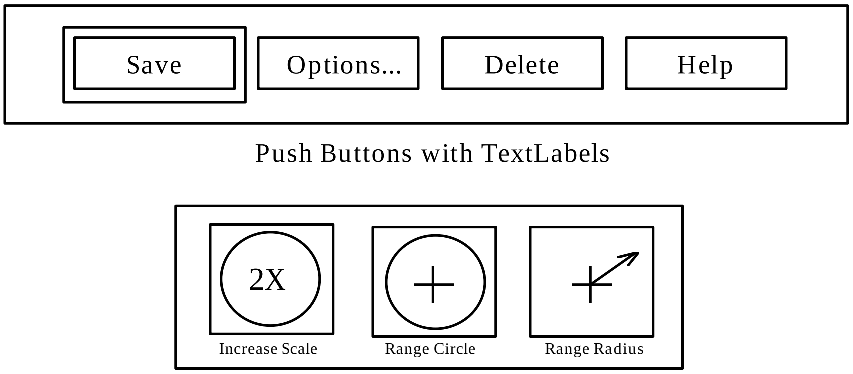
6.1 PUSH BUTTONS

6.1.1 Appearance

A push button is used to initiate an action. The button contains a text or graphic label, as shown in figure 6-1, indicating the action executed when the button is activated. The label is centered, and enough space is provided between it and the rectangle surrounding it so as not to restrict the legibility or visibility of the text or graphic in the push button. The push button designated as the default has an extra border around it.

Motif Only: Push buttons have the default dimensions defined by Motif, except for margin width which is 8 pixels (per Kobara).[1]

Windows Only: Push button height is 22 pixels, when the default font is used (see section 9.1.1).



Push Buttons with TextLabels

Push Buttons with Graphic Labels

Figure 6-1. Example push buttons in Motif.

All of the push buttons in a group are the same size. If the buttons have text labels, the buttons are wide enough to display the longest button label. If the buttons have graphic labels, the buttons are large enough to display the largest graphic. Exceptions to these size guidelines may occur in order to accommodate a button with a label or graphic that is significantly longer or larger than the others in the group, especially when space in a window is limited. The first letter of each word in a text label is capitalized, except for prepositions and articles. The label includes an ellipsis if activating the button results in another window (other than a help window or a message window to confirm the action executed by the push button) being displayed.

The vocabulary in appendix C is used to construct push button labels whenever the application performs the actions listed. New vocabulary can be created as needed to describe actions not in the appendix. New vocabulary is not created for actions already defined, and existing vocabulary is not used to describe new actions. If new vocabulary is created, it is a verb and stated in active voice; it describes the results of pressing the button and reflects the action that will be taken by the application rather than the user. The names of actions are congruent (e.g., Save / Delete, On / Off, In / Out), with the same vocabulary used to describe an action throughout the application. Terms such as “All” are used in labels (e.g., Select All, Delete All) only when there is no ambiguity as to the object or data element to

[1] Resource settings that implement visual design recommendations in Kobara are provided in appendix D.

which “All” refers. If a label may have multiple referents within a window, the name of the object or data element is used instead of “All.”

Action icons (i.e., push buttons with graphic labels) are designed so users can easily identify the function performed when the icon is selected. Each graphic is unambiguous and easily distinguished from the action icons with which it is displayed. The graphics for action icons that represent opposite functions (e.g., Save, Delete) are designed to mirror each other. Action icons do not contain an ellipsis, even if activating the push button with the graphic displays another window.[2] Section 8.1.2.8 describes the design of action icons in toolbars.

6.1.2 Behavior

Placing the pointer on a push button and clicking BSelect activates a push button. When the location cursor is on a push button, <Space> (or <Select> in Motif) activates it from the keyboard. When a push button is activated, it highlights (i.e., changes color) and the action represented by the push button is executed. Releasing BSelect outside the push button does not activate the push button. <Enter> (or <Return> in Motif) activates the default push button in a window.

6.2 RADIO BUTTONS

6.2.1 Appearance

Radio buttons are used in groups to select one option from a set of mutually exclusive options. A radio button consists of a circular indicator, followed by a label describing the option represented by the button, as shown in figure 6-2.[3]

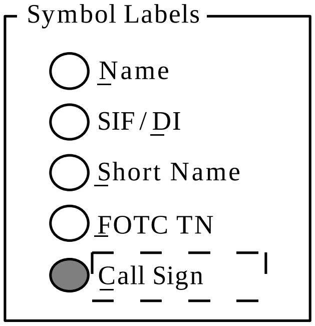
Motif Only: Radio buttons have the default dimensions defined by Motif (per Kobara).

The radio button label defines the state being set by the user. The first letter of each word in the label is capitalized, except for prepositions and articles. If a radio button cannot be selected, its label is grayed out to indicate its unavailability.

> Motif Only: When a group of radio buttons is displayed, one of the buttons is selected. If users need to select none of the options in the group, a radio button labeled None is provided as an option, rather than allowing users to deselect all of the buttons in the group.
>
> Windows Only: Radio buttons can be used to represent a fixed or heterogeneous set of attributes for selection. In the former case, the radio button corresponding to the current attribute is selected when the buttons are displayed. In the latter case, all of the buttons in the group are deselected when displayed, and choosing any button applies the option to the entire selection.

[2] A set of action icons for frequently executed actions will be included in a future version of this document.

[3] In previous versions of Motif, radio buttons had diamond-shaped, rather than circular, indicators.



Symbol Labels

○ Name
○ SIF / DI
○ Short Name
○ FOTC TN
● Call Sign

Figure 6-2. Example radio buttons in Windows.

6.2.2 Behavior

Placing the pointer on a radio button or its label and clicking BSelect selects the button.

Motif Only: When focus is on a radio button, pressing <Space> or <Select> selects the button.

Windows Only: Moving the location cursor to a radio button selects it from the keyboard.

When a radio button is selected, it highlights and any previously selected button in the group is deselected. Only the select state of the option changes; selecting a radio button does not initiate an action or display a dialog window. If the radio button is in a window with a default action, pressing <Enter> (or <Return> in Motif) in a radio button selects the button and executes the action.

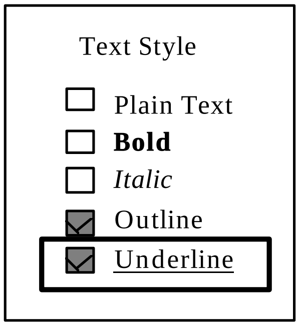
6.3 CHECK BUTTONS

6.3.1 Appearance

Check buttons are used singly or in groups to indicate a nonexclusive setting; selecting a check button toggles to the setting or state indicated by the label. A check button consists of a square-shaped indicator, followed by a label describing the option represented by the button, as shown in figure 6-3.

Motif Only: Check buttons have the default dimensions defined by Motif (per Kobara).

The check button label defines the state being set by the user. The first letter of each word in the label is capitalized, except for prepositions and articles. If a check button is unavailable for selection, its label is grayed out to indicate its unavailability. A check button (rather than two radio buttons) is used if an option can only be set to on or off.



Text Style

□ Plain Text
□ **Bold**
□ *Italic*
☑ Outline
☑ Underline

Figure 6-3. Example check buttons in Motif.

Windows Only: Check buttons that are used to set the properties of a selection can have three states: on, off, or indeterminate. A check button is indeterminate when the selection is heterogeneous; for example, a font style check button is indeterminate when the selected text contains both normal and bold styles. When the state of the check button is indeterminate, the indicator is filled with a gray pattern. Clicking on the check button cycles through the three states, with the appearance of the indicator reflecting the current state.

6.3.2 Behavior

Placing the pointer on a check button or its label and clicking BSelect selects the button. When focus is on a check button, <Space> (or <Select> in Motif) selects the button from the keyboard. When a check button is selected, the indicator highlights and contains a check mark[4] in Motif or contains an X in Windows; any previously selected check button in the group remains selected. When users select a check button, only the select state of the option changes; selecting a check button does not initiate an action or display a dialog window. If the check button is in a window with a default action, pressing <Enter> (or <Return> in Motif) in a check button selects/deselects the button and executes the action.

6.4 LABELS

A label is used to display static text and graphics in a window. Static text (e.g., titles, headings, and directions) is presented in a label and not in a text field. A label is not selectable, and it is not traversable from the keyboard. The first letter of each word in the label is capitalized, except for prepositions and articles.

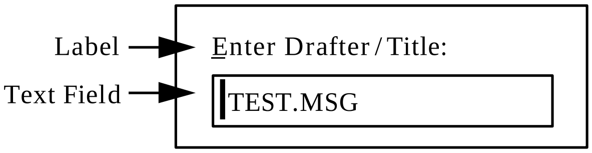
Motif Only: Labels have the default dimensions defined by Motif, except for margin height which is 6 pixels so that labels align properly with other controls (per Kobara).

6.5 TEXT FIELDS

6.5.1 Appearance

A text field, shown in figure 6-4, is used to enter and edit text. A text field includes a label describing what is to be entered in the field. The label is placed either to the left or above the field and has the same background color as the window in which it appears. The label is followed by a colon, and the space between the colon and the text field is empty (i.e., does not include delimiters or underscores). The label is grayed out if the text field is unavailable for text entry.

Motif Only: Text fields have the default dimensions defined by Motif, except for margin height which is 2 pixels in order to be consistent with the appearance of other controls (per Kobara).



Enter Drafter/Title:

TEST.MSG

Figure 6-4. Example text field and label in Windows.

Cues regarding the format of the text to be entered in the text field are presented in the message bar of the window if one is present; otherwise, the label provides these cues. If a unit of measurement (e.g., feet, miles) is always associated with a field, it is displayed as part of the label, as shown in figure 6-5,

[4] Previous versions of Motif indicated the select state of a check button only with highlighting.

and does not have to be entered by users. Cues regarding whether text entry is mandatory or optional are presented in the message bar if one is present; otherwise, these cues are provided in the label or color coded in the text field itself. The label is worded to be clearly different from the labels used in other text fields in the window. If the labels in a group of text fields are highly redundant (e.g., ship name, ship UIC, ship homeport), the common word is removed from the label (e.g., Name, UIC, Homeport) and placed instead in the heading (e.g., Ship Information) that describes the group of fields.

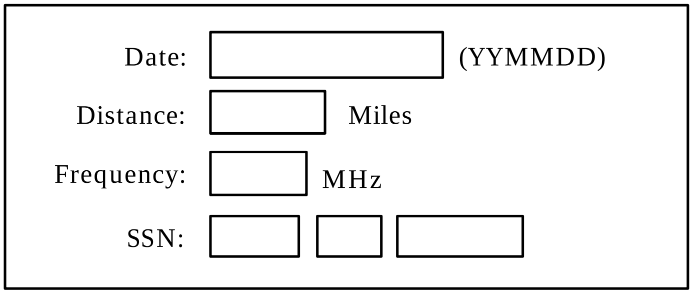


Figure 6-5. Example text field labels providing format cues.

The text entry area is long enough for users to enter the information required. If the information is a fixed length, then the field is the same length as the information. If the information varies in length, the field is at least as long as the longest information. When window space is limited, the text field includes scroll bars if the information is longer than the field or extends beyond a single line. Strings of characters (that are not a word) longer than five to seven characters are entered in smaller chunks. The field format is meaningful to users (e.g., year, month, and day in a date; the three parts of a Social Security number) and consistent with their expectations. Routine or default data, data already known by the application, or data that can be computed by the application is automatically entered in a field whenever possible. For example, if fields are provided for the start date, end date, and duration of a mission, users enter only two of the values and the application calculates the third.

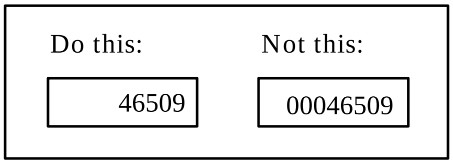
These specifications call for editable text to be displayed in a text field and noneditable text to be displayed in a label. However, in some cases, the application may need to present application-generated data that is not editable in a text field. If dynamic noneditable text is displayed in a text field, the field has a different appearance (e.g., background color) than an editable text field.
When the pointer is on a noneditable field, its shape does not change to an I-beam. A noneditable text field is not selectable; i.e., clicking on it does not change its appearance or display a text cursor in the field.

6.5.2 Behavior

6.5.2.1 Supporting Text Entry and Manipulation

Section 2.2.3 describes text entry modes and actions, and section 3.2.2.4 addresses text cursor behavior during navigation.

When variable-length information is entered in a text field, the information is automatically justified or truncated; users do not have to enter leading characters to fill the space available (see figure 6-6). Text characters are displayed as typed by users (with the same capitalization and punctuation) unless a special format is required (e.g., composing a message). When the application presents stored text in a text field, it is displayed in a consistent format so users can recognize it as such, and any text editing by users is converted into this format.



Do this: 46509

Not this: 00046509

Figure 6-6. Example of automatic justification during text entry.

Users can enter numeric data from either the keyboard or the numeric keypad. The amount of data that has to be entered in a text field is minimized; for example, users are able to enter an abbreviation rather than an entire word and do not have to enter the unit of measurement associated with a number value. The application performs automatic entry of data into a text field (e.g., prefills a date/time group field with the current date and time, or a lat/long field with the current position of the user's ship). In addition, if the application supports position hooking (i.e., allow users to click on a geographic location and have the coordinates of the position entered into a lat/long field), it provides an indicator in the windows where this feature is available (e.g., in the label for the field, with a special symbol next to the field).

Users are not required to enter data in a mandatory field before moving to another field. Users can accept the default value in a text field by tabbing to next field in a window; tabbing does not affect the default. If users modify the default but do not save the change, the change does not affect the default when the window containing the field is displayed again.

Autotabbing is available in text fields only when data such as date, time, latitude, and longitude are broken into smaller groups of characters, with each group entered in a separate text field (e.g. in the SSN field in figure 6-5). In this case, autotabbing can be used since users consider the characters to be a single data value and expect to enter the data without the need to tab between the fields; while separate text fields are intended to improve readability and minimize the opportunity for error, they should not interfere with efficient data entry by users.

6.5.2.2 Error Checking and Correction

The application performs error checking on the data entered and provides feedback when errors are detected. Users can fix errors by editing individual characters in the field, rather than having to erase and retype the entire field.

When users enter data in fields that are unrelated to one another, they receive feedback about an error (e.g., data in an incorrect format or outside range requirements) when they tab out of the text field but they are not prevented from leaving the field. Error feedback is provided in the message bar of the window if one is present; otherwise, feedback is provided by changing the appearance of the field with the error and/or in a separate message window. When users enter data in fields that are interdependent (i.e., the data in one field are correct or in error based on data entered in other fields), they receive feedback (e.g., in the message bar or a message window) concerning the errors when they attempt to accept or save the data.

Users can save the data they have entered at any time by executing an explicit action (e.g., activating a Save, Apply, or OK push button). The application informs them if any errors are present and does not accept (i.e., save) the data until all errors are corrected. The application provides feedback (e.g., a confirmation message) to indicate the data have been saved. When users enter data, all are saved, regardless of text cursor position in the window.

6.6 LIST BOXES

6.6.1 Appearance

A list box such as the one shown in figure 6-7 is used to select one or more items from a set of items. The items in the list are displayed vertically, with one item per line. A vertical scroll bar appears to the right of the items in the list when the number of items exceeds the space available. If the list box includes a title, it is a label describing the purpose or contents of the list; the label appears above the list and is not followed by a colon.

Motif Only: Lists have the default dimensions defined by Motif, except for margin height and width, both of which are 2 pixels in order to provide space between the list borders and contents (per Kobara). Kobara also recommends that a vertical scroll bar be displayed even when the list is short enough to fit in the display area.

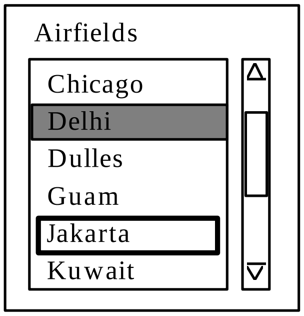


Figure 6-7. Example list box in Motif.

Windows Only: If a list is inactive (i.e., unavailable), its title is dimmed. If an item in a list is unavailable for selection, it is omitted from the list (rather than displayed as unavailable). If scrolling is not possible (e.g., the list is inactive, all of the items are visible), the stepper arrows are dimmed, the slider is removed, and the color of the trough region is changed to that of the window background.

The size of a list box depends on the amount of space available in the window in which it is displayed. A list is large enough to display six to eight items at a time, or all of the items if there are fewer than six. A list is wide enough so that users can read all of the items without scrolling horizontally; if items differ significantly in length, the list is wide enough to display the items of average length and includes a horizontal scroll bar to allow users to read the longer items.

Motif Only: A horizontal scroll bar appears automatically next to the list, rather than increasing the display area, when the width of the list grows beyond that area (per Kobara).

If the items in a list represent possible attribute values for a selection, the current value is selected (and highlighted) when the list is first displayed. If the selection is heterogeneous, none of the items is selected when the list is displayed.

The items in a list are presented in sequential order based on the nature of the items and the sequence in which users expect the items to occur (e.g., chronological, alphabetical, sequential, functional, by importance). For example, a list of port names is ordered alphabetically, and a list of messages by precedence, date-time group, or a combination of the two (e.g., date-time group within precedence). Selecting an item in a list does not affect the order of the items. When items are added to a list, they appear in their correct position within the list (e.g., in numerical or alphabetical order) rather than at the end of the list. Whenever the content of a list is updated through an automatic process, the list does not scroll automatically to the item that was added; the list scrolls only in response to a user action (e.g., using a scroll bar).

6.6.1.1 Drop-Down List Boxes (Windows Only)

A drop-down list is used instead of a regular list box when the space available in a window is limited. A drop-down list consists of a noneditable text area showing the currently selected item, a down-pointing arrow button, and a list of items that is displayed when the arrow button is depressed. The same guidelines concerning height and width apply to an open drop-down list as to a regular list. The arrow button abuts the text area in drop-down list box (in contrast to a drop-down combo box where there is a small gap between the field and the arrow button). When users select an item in the list, it is displayed in the field and the list closes.

6.6.1.2 Multiple Selection List Boxes (Windows (Only)

Windows distinguishes between list boxes supporting single selection and those supporting multiple selection. A multiple selection list is used when users need to select multiple discontiguous items. This type of list includes a check button preceding each item (as a visual cue that multiple items can be selected). When users select an item in the list, an X appears in the check button for that item.

6.6.2 Behavior

6.6.2.1 Navigation and Selection

Placing the pointer on a list item and clicking BSelect moves the location cursor to the item and selects it. If other pointing device selection methods are available in the list, they are performed as defined in table 3-1. If the window containing the list has a default action, double clicking on an item selects the item and executes the action.

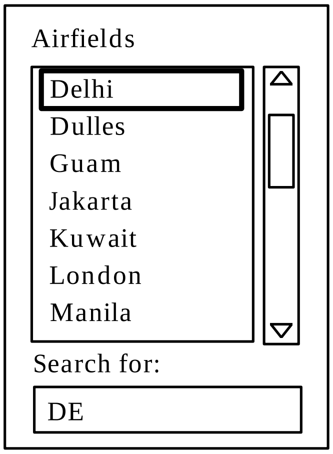
<Up> and <Down> move the location cursor to the previous and next items in a list; <Left> and <Right> scroll the list one character to the left and right. <PageUp> and <PageDown> move the location cursor to the item one page up and down in the list; <Ctrl><PageUp> and <Ctrl><PageDown> scroll the list one page to the left and right. <Home> and <End> scroll to the leftmost and rightmost edge of the list; <Ctrl><Home> and <Ctrl><End> move the location cursor to the first and last items in the list. Selection is performed from the keyboard as defined in table 3-2 for Motif applications and in table 3-3 for Windows applications.

6.6.2.2 Speed and Incremental Search in Lists[5]

A speed search capability is available in lists containing more than 10 items. When the list has focus and users type a character, the list scrolls to the first instance of an item that begins with that letter, and the location cursor moves to that item. In Windows, the item is also selected. When users type the character again, the list scrolls (as needed) and the location cursor moves to the next item that starts with the character.

An incremental search capability is available in lists containing more than 50 items. In this case, a text field is provided with the list, as shown in figure 6-8. Users can type the first few letters of the item desired in the text field or enter wild card characters to search for specific text patterns, as described in section 9.1.5. When they press <Enter> (or <Return> in Motif), the list scrolls to the first occurrence of an item that matches the letters. Users can scroll through this part of the list to locate the item desired or type additional characters to narrow the search further.

[5] Neither of these capabilities is not currently supported by Motif. However, implementation of a search capability is recommended, especially when users are presented with lists that contain a large number of items. A speed search capability is included in Windows.



Search for: DE

Figure 6-8. Example list and text field used in an incremental search in Motif.

In both speed search and incremental search, if the character(s) typed do not match any of the items in the list, the location cursor does not move, and users receive feedback (e.g., an auditory signal, a message in the message bar, or a message window) to indicate that no match was found. Both search and incremental search are not case-sensitive; if the search has to be case-sensitive, then this information is provided to users (e.g., in the message bar of the window).

6.7 SCROLL BARS

6.7.1 Appearance

Scroll bars are used to view textual or graphic information when it exceeds the space available to display it. Vertical scroll bars control backward and forward movement through the information; horizontal scroll bars control left and right movement.

> Motif Only: Scroll bars have the default dimensions defined by Motif, except that vertical scroll bars have a width of 16 pixels and horizontal scroll bars a height of 16 pixels (per Kobara).

A scroll bar contains a trough region, a slider, and stepper arrows, as shown in Figure 6-9. The trough region is the background of the scroll bar and represents visually the length of the information that users can scroll. The slider represents the window through which users view the information. The relative position of the slider represents the relative position of the information currently displayed in the window. The stepper arrows scroll incrementally through the information and indicate the direction of the scrolling movement. Users can scroll to the top or the bottom of the information but not beyond.

> Windows Only: When a window cannot be scrolled any further, the associated stepper arrow is dimmed.

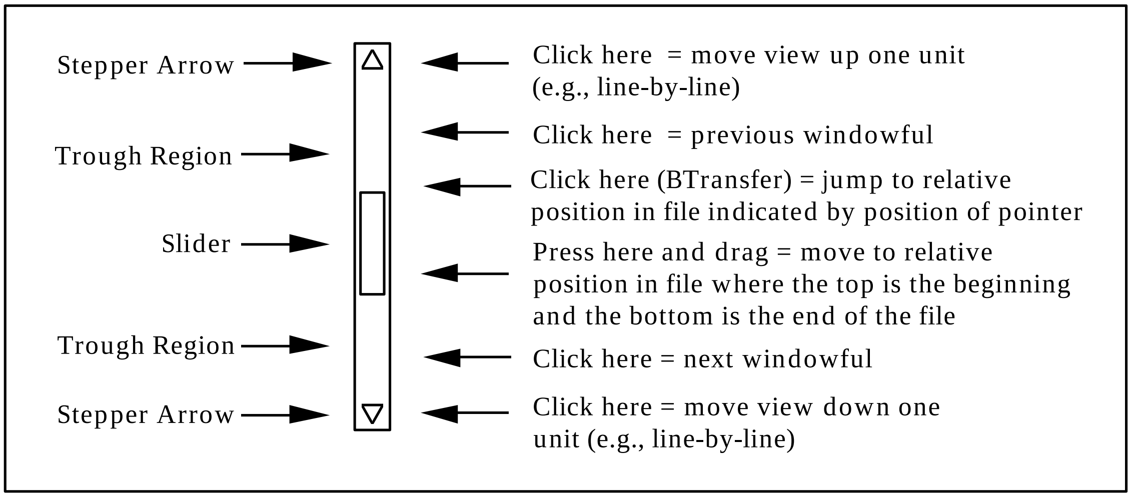

Stepper Arrow
Trough Region
Slider
Trough Region
Stepper Arrow
Click here = move view up one unit (e.g., line-by-line)
Click here = previous windowful
Click here (BTransfer) = jump to relative position in file indicated by position of pointer
Press here and drag = move to relative position in file where the top is the beginning and the bottom is the end of the file
Click here = next windowful
Click here = move view down one unit (e.g., line-by-line)

Figure 6-9. Components of a scroll bar.

6.7.2 Behavior

Pressing BSelect on a stepper arrow moves the slider in one-unit increments (e.g., one line or column) in the direction indicated by the arrow. Pressing BSelect on the trough region moves the slider one window length (or width) minus one unit (for overlap) at a time. Pressing BSelect and dragging the slider moves the slider in the pointer direction. Pressing BTransfer in the trough region moves the slider to the pointer position and then moves the slider in the pointer direction. <Esc> (or <Cancel> in Motif) returns the slider to its position before the sliding operation began.

When the scroll bar has focus, the arrow keys move the slider one increment in the arrow direction. <Ctrl> in combination with the arrow keys move the slider one large increment in the arrow direction. <PageUp>, <PageDown>, <Ctrl><PageUp>, and <Ctrl><PageDown> page in the specified direction. <Home> or <Ctrl><Home> and <End> or <Ctrl><End> scroll to the beginning and end of the scrollable region.

6.8 SCALES AND GAUGES

6.8.1 Appearance

A scale (as shown in figure 6-10) is used to select a value in a range. A scale consists of a trough region, a slider for selecting a scale value, and a label above or next to the slider showing the value currently set. A scale can also have arrow buttons which are used to select a scale value. The scale bar includes tick marks representing the range of available values and is labeled with the minimum and maximum values for the scale.

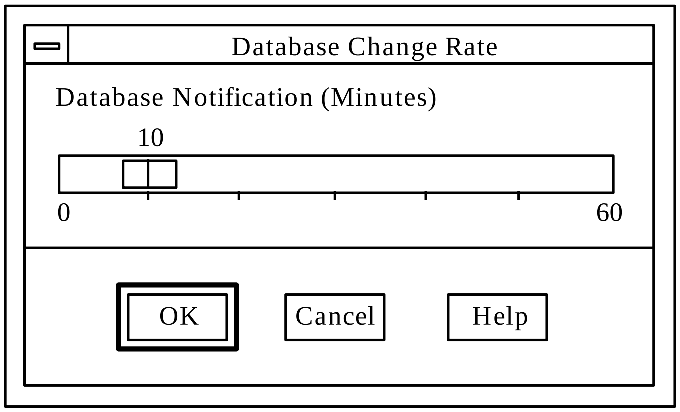
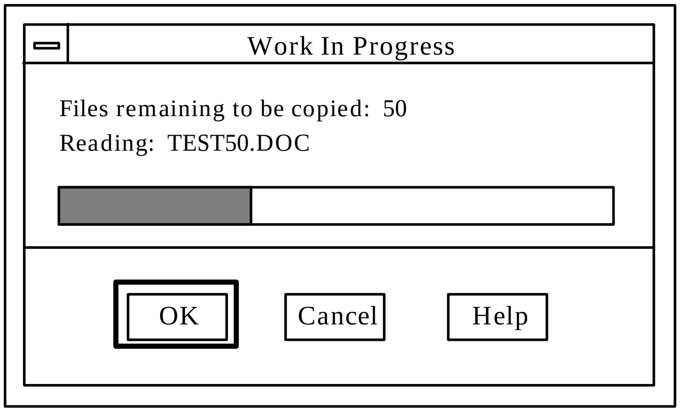


Figure 6-10. Example scale in Motif.

A gauge (shown in figure 6-11) is a display-only version of a scale for presenting values that users cannot change. For example, a working message window can include a gauge to provide dynamic feedback to users on the percent of a process that is complete. If a gauge is used to indicate processing, the trough region fills dynamically to indicate the relative amount of processing completed. If the exact percentage is important, the gauge includes a label indicating the current percentage value, and both the trough and label are updated dynamically. Because the values displayed by a gauge cannot be changed, a gauge does not include a slider or arrow buttons. However, a gauge is able to receive keyboard focus so that users can access Help for that control.



Files remaining to be copied: 50
Reading: TEST50.DOC

Figure 6-11. Example gauge in Motif.

6.8.2 Behavior

If the scale has stepper arrows, pressing BSelect on an arrow moves the slider one unit at a time in the arrow direction. Pressing BSelect on the trough region moves the slider one large increment (defined by the tick marks, if provided) at a time in the direction indicated. Pressing BSelect and dragging the slider moves the slider in the pointer direction. Pressing BTransfer in the trough region moves the slider to the pointer position and then moves the slider in the pointer direction. <Esc> (or <Cancel> in Motif) returns the slider to its position before the sliding operation began.

When the scale has focus, the arrow keys on the keyboard move the slider one increment in the specified direction. <Ctrl> in combination with the arrow keys move the slider one large increment, and <Home> or <Ctrl><Home> and <End> or <Ctrl><End> move the slider to the minimum and maximum scale values.

6.9 SEPARATORS

A separator is used to delimit the elements in a menu (e.g., groups of related options) or a window (e.g., the action and control areas). A separator does not support internal navigation.

Motif Only: A separator extends to the edges of the area it is delimiting (e.g., to the frame border in a window) and has an "etched in" appearance (per Kobara).

6.10 COMBINATION CONTROLS[6]

6.10.1 Combo Boxes

A combo box consists of a text field and a list box displayed immediately below the text field, as shown in figure 6-12. A drop-down combo box (see figure 6-13) is a type of combo box consisting of a text field, a down-pointing arrow button, and a list box that is displayed when the arrow button is depressed. In both cases, users can either select one of the items from the list to display in the text field, or type directly in the field. When users select an item from the list, it replaces any text in the field. When users type in the text field, the text entered does not have to match one of the items in the list.

Windows Only: As users type in the text field, the list scrolls to the first item that begins with the characters being typed.

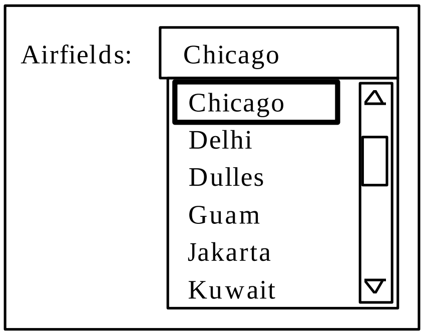


Figure 6-12. Example combo box in Motif.

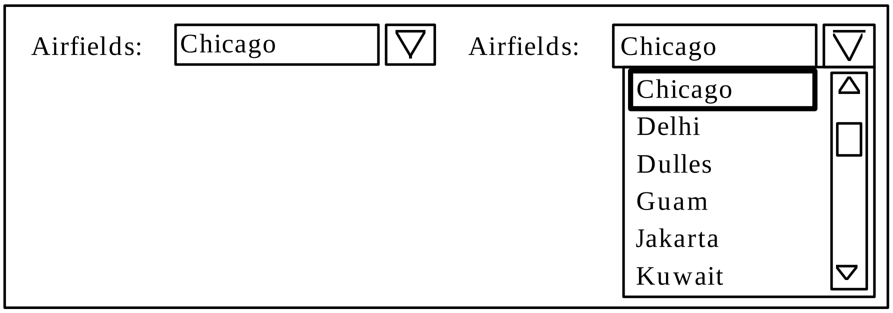


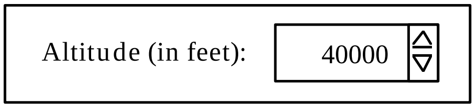
Figure 6-13. Example drop-down combo box in Motif.

[6] Combination controls are not available in previous versions of Motif. Windows distinguishes between combo boxes and drop-down combo boxes while CDE refers only to combo boxes. In addition, a combo box in CDE is the same as a drop-down combo box in Windows.

The list in a combo or drop-down combo box is large enough to display six to eight items at a time, or all of the items if there are fewer than six. A vertical scroll bar is provided when the list is too long for all of the items to be visible. In a list that can be scrolled, the location cursor does not wrap. The combo box is wide enough that users can read all of the items in the list, with the text field the same width as the list. List items appear in sequential order based on nature of items and sequence expected. If it is appropriate for users to make no choice in a combo box, the list includes a "No Choice" item that, when selected, removes any text that is displayed in the text field. When a combo box is initially displayed, the text field can either be empty or pre-filled with a default item from the list. In the latter case, when the combo box receives keyboard focus, the default entry is highlighted so that text typed by users overwrites this text.

6.10.2 Spin Buttons

A spin button is used when users have to enter no more than 20 discrete, ordered values. The values can be numeric or a list of text strings (e.g., months of the year). A spin button consists of a single-line text field, with up- and down-pointing arrow buttons to the right of the field, as shown in figure 6-14. The text field can be editable (e.g., if the list of "spin" entries does not include all possible values) or noneditable (e.g., if the list is short and includes all possible values). When a spin button is displayed, the text field contains a default value. Users click on the up and down arrows to step through the values in the text field, and press on the arrows to step through the values continuously in the direction of the arrow. In both cases, when the largest or smallest value is reached, the entries wrap so that users can cycle continuously through the range of values. If the text field is editable, users can also type a value directly in the field.



Altitude (in feet): 40000

Figure 6-14. Example spin button in Motif.

If spin buttons are used to enter data such as date/time group or lat/long, separate spin buttons are used for each part of the entry. Spin buttons can be combined with standard text fields for data entry (e.g., separate spin buttons for month and day, with a text field for typing the year). If a value is typed in the text field, it is validated for correct syntax and format when users move focus out of the field.

6.11 STANDARD AND NONSTANDARD CONTROLS

6.11.1 Consistent Appearance and Behavior

If the application uses a nonstandard control, it has as much of the standard "look and feel" as possible. These features include a 3-D appearance for controls that can be manipulated by the user and a similar color and shape coding scheme as other controls. If a control is a composite of existing controls, the composite has the appearance and behavior of the component controls.

All of the controls (both standard and nonstandard) in the application are identifiable solely on the basis of their appearance, and all controls with the same function have the same appearance. Users can distinguish between controls that are similar in shape (e.g., a push button and an option menu) on the basis of distinctive visual cues (e.g., an option menu should include a bar graphic). Users do not have to select a control in order to determine what it is and how it behaves.

6.11.2 Adapting Controls When Using Commercial Software

If the application uses commercial off-the-shelf (COTS) software, it configures these products to be compliant with the specifications presented here insofar as possible. In particular, existing controls are not modified in ways that conflict with the appearance and behavior defined in these specifications. For example, if a COTS product does not include a control such as a push button, the application does not include an object with the appearance of a text field and the behavior of a push button in order to provide this functionality. Likewise, the pull-down menus in a window are not altered so that the menu titles function as push buttons; this type of modification is inconsistent with both the normal behavior of menu titles (i.e., when selected, a pull-down menu is displayed) and with Motif and Windows guidelines on the arrangement of objects in windows (i.e., push buttons are not placed in a menu bar). When selecting a COTS product, consideration needs to be given to the extent to which the product can be adapted to fit the style defined by these specifications. If significant discrepancies (such as those described above) will result, the product may be inappropriate for use given the inconsistencies in “look and feel” that will result if it is integrated with other style-compliant applications.